

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Subhashini Subramaniam	Art Unit: 3623
Appl. No.: 10/722,408	
Filed: 11/28/2003	Examiner: CHOI, PETER H
For: Meta Directory Server Providing Users the Ability to Customize Work-flows	Attorney Docket No.: SUN-007/030215

Reply Brief Under 37 CFR § 41.41

Mail Stop **Appeal Brief - Patents**

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Further to the Appeal Brief Dated 29-March-2010 ("Appeal Brief") and as a response to the Examiner's Answer dated 07-July-2010 (hereafter "Examiner's Answer"), Appellants submit this reply brief under 37 CFR § 41.41.

Appellants address some of the points made by the Examiner in a section entitled, "(10) Response to Argument" in pages 17-23 of the Examiner's Answer.

10 Starting 2nd paragraph on page 18 of the Examiner's Answer, it is stated that:

Figure 9 demonstrates that the sequence of Built-in Tasks 910-919 are not edited, but allows that each Built-in Task may correspond to an extension point, and each extension point might be associated with a custom task. The Examiner asserts that by allowing the extension point of a Built-in Task to be associated with any of a plurality of custom tasks constitutes constructing a customized, modified workflow sequence. Although the custom task itself or the sequence of Built-in Tasks may not have been edited or modified in any way, the Examiner asserts that the modularity of the custom tasks and how they may be interchangeably used with each extension point in each Built-in Task Examiner results in the work flow being edited. For this reason, the claimed "not edited" feature is not inherent to the disclosure of the instant application. For example, for Built-in Task 1, the

extension point may be associated with Job 1, or Job 2, or Job 3, which is executed before the workflow returns to Built-in Task 2. The Examiner asserts that a workflow comprising of performing Job 1 within Built-in Task 1 is different than a workflow performing Job 3 within Built-in Task 1; the workflow has been modified to change the association of extension point of Built-in Task 1 from Job 1 to Job 3, therefore the workflow has indeed been edited. Further, the Examiner maintains the position that the specification does not describe in such a way as to reasonably convey to one skilled in the relevant art that the inventors at the time the application was filed, had possession of the step of specifying a custom task associated with an extension point in a work flow "without editing said work flow". (Emphasis Added)

Paragraph 0085 of the published (subject) application clearly states that built-in tasks 910-919 of Figure 9 together represent a work-flow sought to be customized. EP1-EP4 (of 930) there provide the extension points such that custom tasks CT1-CT4 (of 950) are selected for execution when corresponding associated extension point is reached. Due to such operation, the work flow defining the built-in tasks 910-919 is not edited, even though different custom tasks are associated to the same extension point. The Honorable Board is accordingly respectfully requested to reverse the rejection under 35 U.S.C. § 112.

25

Starting 3rd paragraph on Page 19 of the Examiner's Answer, it is stated that:

Chatterjee allows a work flow sequence of tasks to be created and executed. The claimed method requires for a sequence of built-in tasks to be provided, wherein said sequence implements a workflow. Chatterjee provides users with the ability to create new workflows [Column 5, lines 51-61], as well as allowing users to assign specific operations, workbaskets and so on to a workflow step [Column 6, lines 22-26]. Chatterjee also allows insertion of a sub-workflow or complex operation into a current workflow diagram, simplifying design of future workflows by permitting re-use of such sub-workflows and complex operations [Column 7, lines 8-13]. Although Chatterjee provides users with the ability to modify existing workflows, Chatterjee also provides the ability to create a new workflow so that it has custom tasks or custom predefined sub-workflows associated with each workflow tasks where the custom tasks and predefined sub-workflows are modular components that may be used with a task within the workflow sequence of tasks, wherein the custom tasks and predefined sub-workflows have not been edited. The

5 Examiner asserts that there is a difference between defining a new workflow and editing a workflow, namely that editing requires a workflow to have already been defined. In short, Chatterjee allows editing of a workflow, but in no way whatsoever requires editing of a workflow as alleged by the Appellant. Thus, the Examiner asserts that a workflow in Chatterjee may simply have specific operations or workbaskets assigned to a workflow step, whilst the sequence of workflow tasks remains unedited. (Emphasis in 10 Original, Emphasis Added)

15 It is respectfully pointed out Chatterjee requires editing the workflows if customization is to be effected to the workflow. In other words, the only scenario when the sequence of workflows of Chatterjee remain unedited (as alleged by the Examiner above), is when there is no further changes or customization sought to be effected in the sequence of Chatterjee.

20 In sharp contrast, the invention of the subject patent application is for customization of a work-flow (by different users) without editing the sequence of built-in tasks (workflow) provided by a designer. The Honorable Board's attention is directed to the fact that the workflow builder display 300 of Figure 3 of Chatterjee is for contiguous construction (akin to an editor).

25 In the last two paragraphs on Page 20 of the Examiner's Answer, it is stated that:
Appellant argues that Bacon does not provide the ability to specify the personal subflow to be executed.
The Examiner respectfully disagrees. Bacon allows each user to instantiate and initialize a defined sequence of personal subflow activities to be performed. The entry point of the personal subflow may not be the starting point of a workflow [Column 9, lines 6-12]. In other words, the personal subflow must be attached or assigned to a corresponding activity within the workflow and not the starting point. The activity which should be next in the personal subflow is specified given a set of existing conditions [Column 9, lines 27-30]. The personal subflow is stored so that it can be used later in defining a workflow, and the workflow defined may include the personal subflow and assigned to a given participant, at which point the actor to which the personal subflow is assigned is then bound with the personal subflow [Column 9, lines 44-50]. When the server is initialized with a workflow definition that contains an activity to be performed by a personal subflow, the work item is eventually routed to the

5 participant who is defined within the workflow as the actor to perform (i.e., execute) the personal subflow [Column 9, lines 52-63]. (Emphasis in Original, Emphasis Added)

The Honorable Board's attention is first directed to the fact that the Examiner relies on Bacon for the claimed feature of, "... wherein each user is provided the ability to specify a corresponding custom task associated with said extension point without editing said work flow, the custom task specified by a user containing corresponding program logic to provide a customization desired by the user.." (Emphasis Added).

10 While Bacon appears to disclose personal sub-flows for each user, the Honorable Board's attention is directed to the fact that it is the developer, who decides such sub-flows and assigns to individual users (see Col. 9 lines 26-38 of Bacon), as also noted in page 11 lines 10-30 of the appeal brief.

15 In the paragraph starting at the end of page 21 of the Examiner's Answer, the Examiner makes several erroneous points. The Honorable Board's attention is first pointed to the relevant claim language:

20 " a built-in task in said sequence of built-in tasks containing an extension point at a point of interest in said work flow for users,

25 " executing said custom task when said extension point is reached during execution of said built in task in an instance of said work-flow for said user; and
30 continuing execution of said sequence of built-in tasks from said extension point in said built-in task after executing said custom task in said instance such that all of said sequence of built-in tasks are executed to complete synchronizing/ consolidating data between said two data sources. (Claim 1, Emphasis Added)

35 The claim recites that execution continues from the extension point in the built-in task from which the custom task was executed (when the extention point was reached). Paragraph 007 ("in the middle of execution of a built-in task") of the published (subject) application, clearly supports such an interpretation.

40 The Examiner appears to not appreciate this express limitation of the claim, as can be observed from the below highlighted portions starting at page 21 last paragraph of the

Examiner's Answer:

The Examiner respectfully disagrees. Chatterjee teaches a "Return to" button that inserts a "return to" step that routes an object to a user who created that object [Column 7, lines 40-43, Figure 4].

The Honorable Board's attention is directed to the fact that the "return to" feature of above routes on object to a user who created it. The disclosure there simply does not pertain to execution flow returning to the point at which the custom task was executed.

10

The Examiner had further stated:

15 ... Each of the buttons the user to insert or perform an action within a workflow, such as linking steps, deleting choices, insertion of a "detour" path, insertion of flow control points, adding insertion points, inserting objects, inserting stop points, inserting distribution points, collection points, broadcast points, workbaskets and sub-workflows or complex operations. As seen from Figure 3 of Chatterjee, execution of the work flow continues from decision point 312 (i.e., extension point) and may include processing of custom tasks/activities. From decision point 312, processing may continue to Accepted PO Path 317, or to Data Entry 313. From Data Entry 313, processing may continue to Decision Point 314. From Decision Point 314, processing may return to Accepted PO Path 317 or continue to Rejected PO 315. Thus, from each of the decision points (i.e., extension points), execution of the work flow continues from the decision point and proceeds to the next step of the work flow sequence. This is deemed to be analogous to the claimed step of returning control to the same point of the work flow execution upon completion of the custom task because once control returns to the built-in task where the custom task is executed via the extension point, the workflow continues to the next step of the work flow sequence of built-in tasks as the task at the preceding built-in task has already been completed. In example, in a simple authorization work flow comprising the sequence of 'Till Out Appropriate Form' and 'Obtain Approval', an extension point at the built-in task of 'Fill Out Appropriate Form' might include the custom task of 'Obtain Form X'. After the custom task of 'Obtain Form X' is completed, control returns to the work flow sequence, but since the built-in task of 'Fill Out Appropriate Form' has already been completed by virtue of completing the custom task at the extension point of said built-in task, control then returns to the next step in the work flow sequence, or 'Obtain Approval'. Although the steps immediately following decision points may be custom tasks/activities, this is not precluded by the claimed

invention.

The Examiner simply ignores the above noted claim feature, in making the assertions of the emphasized portion above. In particular, the Examiner comes to the
5 conclusion that the 'preceding' built-in task (from which custom task has been executed) has already been completed, contrary to the claimed feature.

With respect to claim 4, in the second paragraph of page 23 of the Examiner's Answer, it has been stated that:

10 ... The Examiner notes that the specification does not set forth any special definition for "extension point", therefore the broadest reasonable interpretation has been applied. As recited in the claims, each "extension point" is associated with a custom task and contained within a built-in task of a workflow sequence of built-in tasks. In other words, the "extension points" are similar to a decision node within a flowchart, where they are associated with tasks that initiate a decision, and are associated with tasks that are performed subsequent to the decision being made. Chatterjee allows users to define a plurality of rules using conditional statements that are evaluated at decision points (i.e., extension points), as well as providing a selection of routing choices that will be available as rule results [Column 9, lines 38-46]. In other
15 words, based on the evaluation of the conditional statement rules, only certain routes/paths of the workflow are available for subsequent execution. As the purpose of the claimed extension points is to dictate the custom task to be associated with it within the workflow, the Examiner asserts that the decision points of Chatterjee are indeed an analogous concept.

20 The Honorable Board's attention is pointed to the fact that the 'decision points'
merely imply a choice of paths, but is not an inherent requirement that execution return
25 to such point and continue from there onwards. It is accordingly requested that the Honorable Board find the Examiner's assertion of above to be erroneous.

30 At least for such reasons, the Honorable Board is respectfully requested to find that the art of record does not establish a prima facie case of obviousness and reverse the rejections of record.

Conclusion

The reversal of the Examiner's rejections of claims 1, 3-6, 8-11, 13-16, 18-21, 23-26, 28-31 and 33-36 is respectfully requested. The Office is invited to call the Undersigned Representative at 443-552-7281 (4AM-noon EST, else voicemail) if it is
5 believed that an interview might be useful for any reason. The Director is hereby authorized to charge any underpayment of fees (including extension fees), or credit any overpayments to Deposit Account No.: 20-0674.

Respectfully submitted,
/Narendra Reddy Thappeta/
Signature
Printed Name: Narendra Reddy Thappeta
Attorney for Appellant
Registration Number: 41,416

Date: 07-Sep-2010